UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/918,600	07/30/2001	Ping-Sheng Tseng	16503-302501	8219

54698 7590 08/24/2006

RAYMOND R. MOSER JR., ESQ.
MOSER IP LAW GROUP
1040 BROAD STREET
2ND FLOOR
SHREWSBURY, NJ 07702

EXAMINER	
SAXENA, AKASH	

ART UNIT	PAPER NUMBER
2128	

DATE MAILED: 08/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

AUG 2 4 2006

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/918,600
Filing Date: July 30, 2001
Appellant(s): TSENG ET AL.

Raymond R. Moser, Jr.
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 17th May 2006 appealing from the Office action mailed 2nd March 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is incorrect.

The amendment after final rejection filed on 9th February 2006 has not been entered. Claim 27 was amended to clear language. Issues presented with the claims earlier were not rectified and were briefly pointed out in the Advisory Action dated 2nd March 2006.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is deficient. 37 CFR 41.37(c)(1)(v) requires the summary of claimed subject matter to include: (1) a concise explanation of the subject matter defined in each of the independent claims involved in the appeal, referring to the specification by page and line number, and to the drawing, if any, by reference characters and (2) for each independent claim involved in the appeal and for each dependent claim argued separately, every means plus function and step plus function as permitted by 35 U.S.C. 112, sixth paragraph, must be identified and the structure, material, or acts described in the specification as corresponding to each claimed function must be set forth with reference to the specification by page and line number, and to the drawing, if any, by reference characters.

The brief is deficient because definition of term "Behavior Processor" and its function & composition in terms of hardware and software is not clear. Points below list the deficiencies in the summary of claimed subject matter and its support is provided below.

1. Appellant identifies "Behavior Processor" provides an architecture implementing behavior applications such as monitors, triggers and memory servers used in integrated circuit design verification. It is further stated that the "Behavior Processor" is integrated in reconfigurable computing (RCC) system (having software model) and RCC hardware array (having hardware model). It is still not clear what "Behavior Processor" comprises.

2. The issue is further complicated by appellant's statements, for example "behavioral aspect of users's integrated circuit design and debug session are implemented in the hardware..." and "behavior functions are performed within the hardware model unless the host testbench process is required" with support in specification presented on pages 204-206.

Examiner's interpretation of the invention from these pages is that once a pre-specified condition occurs in hardware, an interrupt is issued from the hardware model using a test bench callback process to initiate a software model routine (software test bench process in RCC system) servicing the interrupt. The interrupt is serviced in software like any hardware-initiated software interrupts. The ***behavior functions (like monitors)*** are then executed in software alone and not in hardware as claimed, in response to the conditional interrupt. The monitors (e.g. $DISPLAY command) used is command in VHDL language to print the state of a hardware unit and the ***$display*** command known to be un-synthesizable in the art.

Mapping the examiner's interpretation to pages 204-206 of the specification: Test bench process is defined to be "axis_tbcall" which is initiated on interrupt ("trigger_signal") signal from hardware and performs the "task_to_execute" [Pg.204 Line 15] in software [Pg.204 Lines 9-11]. The "task_to_perform" is the behavior function (un-synthesizable in hardware) that monitors through a software only command $DISPLAY [Pg.204 Lines 26-29, Pg. 206 Lines 11-12].

Also see Appeal Brief Pg.5 Lines 5-6 stating, "The task is then executed in software in the RCC workstation (3107)." Hence the behavioral aspect (e.g. monitors) of users's integrated circuit design and debug session are ***do not appear to be*** implemented in the hardware, contrary to appellant's claim.

3. Appellant has further attempted to map the claim to the figures to clarify the issue. Examiner does not agree with the claim mapping and appellant's summary related to claims for the reasons below.

3a. The claim 1 mapping, even if assumed too be correct, at least appears to be misleading and incomprehensible as presented in disclosure. For example claim presents "Behavior level function (3109a)" in claim, however it is labeled Behavior Processor (3109a) in Fig.100.

"Behavior Process (3100, 3103)" in claim includes Workstation 3100 and RTL ASIC 3103 in Fig.99. From Summary presented in Appeal Brief Pg. 3, RCC system includes a workstation. Therefore (RCC) Workstation includes test bench 3101 in Fig.99 but in Fig.100 test bench call back process (3109b) is identified to reside outside the RCC 3107. Further RCC 3107 is shown to reside outside the Workstation 3106 in Fig.100 contrary to disclosure of Fig.99.

The "testbench call back process (3109b)" identified in claim is Internal Memory 3109b in Fig.100.

"reprogrammable logic element (3109)" is not labeled in Fig.100, however another block FPGA 3108 is labeled separately.

The numbers do not match the labels in essence and hierarchy of elements is incomprehensible.

3b. Claim 1 discloses "Behavior Processor" limitation in the preamble, which does not limit the claimed invention. Further, even if the "Behavior Processor" is included in the body of the claim, the support provided by the appellant in specification Pg. 204-206 provides definition contrary to limitation that "behavioral functions" are implemented in hardware. Appellant had earlier provided support in specification Pg.192 Lines 2-4, which is merely a statement that "behavioral functions" are implemented in hardware and lacks any embodiment. In view of contrary definition from page 204-206 examiner disagrees with the statement on page 192 for support.

3c. The claim 1 language discloses "a test bench call back process *for responding to the behavior level function in the programmable logic element* by sending a signal to the host test bench process". As shown in step 2 above the behavior level function is not in the programmable logic element. Only the condition that causes the interrupt is in the hardware; it is the interrupt, which initiates a software task on RCC system, to execute the behavior level function in software. Hence the claim language has written description problem and summary appears to be incorrect interpretation of specification.

As shown these inconsistencies in the "summary of the claimed subject matter" section render the summary deficient.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

U.S. Patent No. 5,838,948 Bunza

- IEEE Std 1364-1995; IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language.
- "A 145 MHz user-programmable gate array"; do Valle Simoes, E. et al;Rapid System Prototyping, 1995. Proceedings., Sixth IEEE International Workshop on 7-9 June 1995 Page(s):226 - 232

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-11, 13-15, 17-30, 32-34 & 36 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 5,838,948 issued to Geoffrey J. Bunza (BU '948 hereafter).

Regarding Claim 1

BU '948 teaches a behavior processor system for operating a portion of user design and interfacing it with the host test bench process (BU '948: Fig.5, 6; Col.10 Lines 60-65). BU '948 teaches a reprogrammable logic element (BU '948: Col.9 Lines 8-12) for modeling a hardware model for a portion of the user design that includes behavioral level functions (BU '948: Col.5 Lines 62-67; Col.6 Lines 10-13). Further, BU '948 teaches a test bench call back process as control program and associated circuitry that surrounds the hardware emulation environment that communicates with the software simulator, simulating the other hardware control circuitry on the host test bench (BU '948: Col.13 Lines 55-60; Col.10 Lines 43-46, 60-65; Col.11 Lines 11-16). Further, BU '948 teaches the test bench call back process responding to the behavioral level function in the reprogrammable logic element by sending a signal to the host test bench process (BU '948: Fig. 7; Col.12 Lines 63-67; Col.13 Lines 1-6, 8-13).

Regarding Claim 2 & 3

BU '948 teaches that the behavioral level function includes a condition (BU '948: Col.13 Lines 55-64) and occurrence of these conditions triggers the test bench call back process (BU '948: Col.13 Lines 17-23).

Regarding Claim 5 & 6

BU '948 teaches the signal including an interrupt from the test bench call back process to the host test bench process as an I/O trap (BU '948: Col.13 Lines 8-36; 60-64), initiated by the reprogrammable logic element (as processor emulator).

Regarding Claim 7

BU '948 teaches that signal going from the test bench call back process to host test bench process includes data (BU '948:Col.12 Lines 25-32; 4-6).

Regarding Claim 8

BU '948 teaches that reprogrammable logic element temporarily suspends operation upon the occurrence of a condition (BU '948: Col.15 Lines 8-13).

Regarding Claim 9

BU '948 teaches that reprogrammable logic resumes operation from the point at which operation was temporarily suspended upon service of the signal by the host test bench (BU '948: Col.15 Lines 8-35).

Regarding Claim 10

BU '948 teaches that reprogrammable logic element temporarily pauses operation on occurrence of a condition (BU '948: Col.7 Lines 63-67; Col.15 Lines 8-13).

<u>Regarding Claim 11</u>

BU '948 teaches reprogrammable logic element includes a clock that controls the speed of processing instructions and data in reprogrammable logic element (BU '948: Col.7 Lines 61-67).

<u>Regarding Claim 13</u>

Claim 13 discloses the similar limitations as claim 1 is rejected for the same reasons as claim 1. BU '948 teaches Behavioral processor as software kernel, control program and associate control circuitry surrounding the hardware emulator (BU '948: Col.13 Lines 55-60).

<u>Regarding Claim 14</u>

BU '948 teaches modeling a hardware model for a portion of the user design that includes behavioral aspects of the user design (BU '948: Col.5 Lines 62-67; Col.6 Lines 10-13).

<u>Regarding Claim 15</u>

Claim 15 discloses the similar limitations as claim 2 is rejected for the same reasons as claim 2.

<u>Regarding Claim 17</u>

Claim 17 discloses the similar limitations as claim 1 is rejected for the same reasons as claim 1.

Regarding Claim 18

Claim 18 discloses the similar limitations as claim 3 is rejected for the same reasons as claim 3.

Regarding Claim 19

Claim 19 discloses the similar limitations as claim 8 is rejected for the same reasons as claim 8.

Regarding Claim 20

Claim 20 discloses the similar limitations as claim 9 is rejected for the same reasons as claim 9.

Regarding Claim 21

Claim 21 discloses the similar limitations as claim 10 is rejected for the same reasons as claim 10.

Regarding Claim 22

Claim 22 discloses the similar limitations as claim 8 is rejected for the same reasons as claim 8. Further, BU '948 teaches wait is executed by the reprogrammable logic element (BU '948: Col.7 Lines 63-67) on occurrence of a condition.

Regarding Claim 23

Claim 23 discloses the similar limitations as claim 9 is rejected for the same reasons as claim 9.

Regarding Claim 24

Claim 24 discloses the similar limitations as claim 10 is rejected for the same reasons as claim 10.

Regarding Claim 25

BU '948 teaches behavior processor operates when it receives request for service from the host workstation as target circuitry software simulation model running on the host workstation, generating an interrupt for service from processor emulator (BU '948: Col.12 Lines 19-26).

Regarding Claim 26

BU '948 teaches behavior processor operates when it receives request for service from the reprogrammable logic element as interrupt from the processor emulator (target microprocessor) (BU '948: Col.12 Lines 4-19).

Regarding Claim 27

Claim 27 discloses the similar limitations as claim 1 is rejected for the same reasons as claim 1.

Regarding Claim 28

Claim 28 discloses the similar limitations as claim 9 is rejected for the same reasons as claim 9.

Regarding Claim 29

BU '948 teaches suspending operation until the test bench call back process services the signal (BU '948: Col.15 Lines 8-13).

Regarding Claim 30

Claim 30 discloses the similar limitations as claim 2 is rejected for the same reasons as claim 2.

Regarding Claim 32

Method claim 32 discloses the similar limitations as claim 1 is rejected for the same reasons as claim 1. Further, BU '948 teaches the sending of an interrupt from the test bench call back process to the host/test bench process as an I/O trap (BU '948: Col.13 Lines 8-36; 60-64), initiated by the reprogrammable logic element (as processor emulator).

Regarding Claim 33

Claim 33 discloses the similar limitations as claim 9 is rejected for the same reasons as claim 9.

Regarding Claim 34

BU '948 teaches suspending operation until the test bench call back process services the signal (BU '948: Col.15 Lines 8-13).

Regarding Claim 36

Claim 36 discloses the similar limitations as claim 11 is rejected for the same reasons as claim 11.

Claims 4, 16, 31 & 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,838,948 issued to Geoffrey J. Bunza (BU '948 hereafter), in view of IEEE Std 1364-1995 "IEEE Standard Hardware Description Language Based n the Verilog Hardware Description Language" (IEEE 1364 hereafter).

<u>Regarding Claim 4</u>

BU '948 teachings are disclosed in the claim 2 and the preceding claim 1 rejection above.

BU '948 does not teach that condition includes "if-then" conditional statement implemented in hardware.

IEEE 1364 teaches behavioral state machine model in which the states can be written in "if-then" conditional format (IEEE 1364: Example 2 showing the "if-then" scenario with op-code processing in an ALU/processor). The processor on the most generic form is a state machine.

It would have been obvious to one (e.g. a designer) of ordinary skill in the art at the time the invention was made to apply teachings of **IEEE 1364** to **BU '948** to include an "if-then" conditional statement in the behavioral level function on the reprogrammable logic element. The motivation would have been that the hardware model for the reprogrammable logic element is designed using the hardware description language's like VHDL (BU '948: Col.5 Lines 50-53) and IEEE 1364 teaches Verilog as an industry standard for HDL based design (IEEE 1364: Introduction).

Regarding Claim 16

Claim 16 discloses the similar limitations as claim 4 is rejected for the same reasons as claim 4.

Regarding Claim 31

Claim 31 discloses the similar limitations as claim 4 is rejected for the same reasons as claim 4.

Regarding Claim 35

Claim 35 discloses the similar limitations as claim 4 is rejected for the same reasons as claim 4.

Claims 12 & 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 5,838,948 issued to Geoffrey J. Bunza (BU '948 hereafter), in view of IEEE article "A 145MHz User-Programmable Gate Array" by Eduardo do Valle Simoes et al (ED 1995 hereafter).

Regarding Claim 12

BU '948 teachings are disclosed in the claim 11 and the preceding claim 1 rejection above.

BU '948 does not teach reprogrammable logic element clock running at 20 MHz.

ED 1995 teaches reprogrammable logic element clock running at frequency of 145 MHz (ED 1995: Pg.226 Col.2 Section 1.1 Lines 8-10).

It would have been obvious to one (e.g. a designer) of ordinary skill in the art at the time the invention was made to apply teachings of **ED 1995** to **BU '948** and make the reprogrammable logic element run at any frequency up to 145 MHz. The motivation would have been that **ED 1995** teaches the fast implementation using FPGA matrix (ED 1995: Abstract) using strategy to place logic cell in row. Hence it would have been easy to run the implantation at a lower speeds using XILINX XC300 Family FPGA as well (ED 1995: Pg.232 Col.1 Lines 1-3).

Regarding Claim 37

Claim 37 discloses the similar limitations as claim 12 is rejected for the same reasons as claim 12.

(10) Response to Argument

I. In Response of Claims 29 and 34

Claims 29 and 34 are objected to under 35 USC 112 ¶2nd therefore are petitionable and not subjected to appeals process.

II In Response of Claims 1-11, 13-15, 17-30, 32-34 and 36 under 35 USC 102(b)

Appellant has repeated same issues in various claims a many claims, therefore instead of addressing each claim and repeating examiner's position, issues in each claim are identified. If the following claims repeat an already identified issue, the claim number is appended to the issue list to keep the prosecution clear.

Key to map Claims to Issues

Claim 1: Issues A, B and C
Claims 2-3: Issues A and D
Claims 5-6: Issues A and D
Claim 7: Issue A and E
Claims 8-9: Issue A and F
Claim 10: Issue A, F and G
Claim 11: Issue A and G
Claims 13-14: Issue A, C and H
Claims 15 & 17: Issue H
Claim 18: Issue A, C, F and H
Claims 19-20: Issue A, B, C and F
Claim 21: Issue A, C, G and H
Claim 22: Issue A, C, F, G and H
Claim 23: Issue A, C, F and H
Claim 24: Issue A, C, F, G and H
Claims 25-26: Issue A, C and H
Claim 27: Issue A, B and C
Claim 28: Issue A & F
Claim 29: Issue A, B, C and F
Claim 30: Issue A, B and C
Claim 32: Issue F & G
Claim 33: Issue A, B, C, D and F
Claim 34: Issue A, B, C, D and F
Claim 36: Issue A, B, C and G

Issue A:

Appellant has stated that BU'948 does not teach using a programmable logic element for modeling a portion of a design that includes a behavior function. Further, appellant contends that inherency doctrine is relied upon for the basis of rejection.

Issue argued in claims: 1, 2-3, 5-6, 7, 8-10, 11, 13-14, 18, 19-30, 33, 34 and 36

Examiner's Response:

Firstly, examiner has cited following portions of BU'948 (Col.5 Lines 62-67; Col.6 Lines 10-13; Col.9 Lines 8-12 respectively):

> To achieve this degree of accuracy for a highly complex target microprocessor, functions are often represented with detailed structures, although most hardware simulators allow multiple levels of modeling abstraction from a switch or transistor level model to a **high level behavioral model.**

> As discussed above, the processor functional model 26 and the target circuitry functional model 28 can be specified to **various levels of abstraction by a conventional HDL**.

> Another form of hardware emulator, shown in FIG. 5, is a **hardware circuit emulator 94**. The hardware circuit emulator 94 uses reconfigurable circuitry 96, such as a **field-programmable gate array (FPGA)**, to emulate target circuitry functions including the ASIC or custom IC.

BU'948 clearly teaches that various abstraction levels including the behavioral level model can be put on a programmable logic element (like FPGA).

Secondly, Examiner has made a rejection under 35 USC 102(b) so using inherency doctrine is appropriate. The art of "Behavioral synthesis" is well known in the field of behavioral modeling of circuit elements to be implemented on a programmable logic element like FPGA.

<u>Issue B:</u>

Appellant has argued that even if examiner's basis is correct in using the BU'948 reference, the reference is itself not enabling (at least on Appeal Brief Pg.7 & 9).

<u>Issue argued in claims</u>: 1, 19-20, 29, 30, 33, 34 and 36

<u>Examiner's Response:</u>

Examiner asserts that if the <u>patented reference</u> teaches the limitation, it is enabling [MPEP 706.02(j), 716.07]. According to 35 USC 282, all patents are valid (and presumed enabled). The burden of establishing invalidity (in this enablement) of a patent or any claim thereof shall rest on the party asserting such invalidity.

<u>Issue C:</u>

<u>Firstly</u>, Appellant is arguing the distinction between synthesizable and non-synthesizable behavioral function modeled in hardware.

<u>Secondly</u> related to above, appellant is arguing that BU'948 teaching related to "non-synthesizable behavioral function which cannot be modeled in hardware", does not teach "synthesizable function which cannot be modeled in hardware".

<u>Thirdly</u> related to above, appellant asserts that enablement for "non-synthesizable code elements for placement on an FPGA device" is provided specification Pg. 192 pointing to a statement to that effect.

<u>Issue argued in claims</u>: 1, 13-14, 18-27, 29-30, 33, 34, and 36

<u>Examiner's Response:</u>

Firstly, the claim language does not provide distinction between the "synthesizable" and "non-synthesizable" behavioral functions. Therefore until the claim language is amended to this effect, arguments may be moot. However, to further prosecution and make examiner's position clear clarification is provided below.

Secondly, Examiner asserts that Issue A clarifies teaching for behavioral function modeled in a programmable logic element is present in BU'948.

BU'948 Col.9 Lines 49-52 teaches

> Thus, use of unsynthesizable behavioral or high-level design representations, typical of early stages of design, are precluded by the use of hardware emulators.

Examiner agrees that the disclosure above is silent on "modeling synthesizable behavioral functions in hardware" but clearly shows the teaching of "modeling behavioral functions modeled in hardware" **in general** (Shown in Issue A and known as "*Behavioral Synthesis*" in the this art) and merely excludes "non-synthesizable behavioral functions modeled in hardware". Since the claim does not limit this distinction between the two forms of behavioral functions, BU'948 cannot be said to teach away from the instant invention.

Thirdly, Examiner requested support for "non-synthesizable behavioral functions modeled in hardware" and appellant pointed out to page 192 of the specification. This is merely a statement in the specification, which does not appear to have the enablement. Specification Page 204-206 supports examiner's reasons for lack of enablement. Please see "**Summary of Claimed Subject Matter**" above

for reasons causing lack of enablement relating to “modeling non-synthesizable behavioral function in hardware”.

Issue D:

Appellant argues that BU'948 does not teach a *behavior level function having a condition*, while agreeing that BU'948 teaches a "condition" and communication between the processor emulator and hardware simulator. Further asserting that BU'948 cannot teach the condition triggers [and] test callback process and host test bench process.

Issue argued in claims: 2-3, 5-6, 33 and 34

Examiner's Response:

Issue A shows that behavioral functions modeled in programmable logic element (hardware) which is shown as processor emulator by BU'948 (BU'948: Fig.5-6).

BU'948 Col.13 Lines 55-64, 17-23 states:

> The processor emulator 202 contains both control circuitry 204 and a control program, both residing in the memory 226, which can identify conditions of the executing target program 22 and the surrounding hardware environment which will require communication with the hardware simulator 210. These conditions include explicit memory references outside of the loaded target program address range, input/output (I/O) operations, ***interrupt handling***, and instructions dealing with explicit hardware functions, such as RESET. Those skilled in the art will recognize that other instruction may also require such interaction with the hardware simulator 206 The control circuitry 204 in the processor emulator 202 provides hardware capability to detect events requiring the interaction of the target microprocessor and the target circuitry. Similarly, the control program residing in the memory 226 provides software capability to detect events requiring the interaction of the target microprocessor and the target circuitry.
>
> For example, if the event requiring interaction between the processor emulator 202 and the hardware simulator 206 is a 32-bit I/O write instruction (function 16 in FIG. 7), **the processor emulator generates an I/O trap** and the translator 214 fields the trap and determines the type of trap.

The disclosure clearly teaches detection of condition (condition trigger) in hardware and calling (test call back process) the hardware simulator (software) to interrupt the host test bench process (running on hardware simulator).

<u>Issue E:</u>

Appellant argues that there is no teaching of data flowing between two processes, i.e. test bench callback process and the host test bench process.

<u>Issue argued in claims</u>: 7

<u>Examiner's Response:</u>

Examiner respectfully disagrees, the disclosure for Issue D clearly indicates flow of data between the processor emulator (hardware) and hardware simulator (software). Also as indicated above, test bench callback process are the initiated from hardware and the host test bench process are responses from software running on workstation. BU'948 Col.12 Lines 25-32 & 4-6 states:

> However, it should be understood that not all calls for interaction between the target microprocessor and the target circuitry are initiated by the target microprocessor. For example, the target circuitry may generate an interrupt to the target microprocessor, which necessitates the transfer of data from the hardware simulator 206 to the processor emulator 202. The mapper 216 and translator 214 operate in both directions to permit complete interaction between the processor emulator 202 and the hardware simulator 206. <u>For data going from the processor emulator 202 to the hardware simulator 206, the translator 214 determines the nature of the event and translates from the first data format to the intermediate data format and the mapper 216 maps the data in the intermediate data format to the second data format and generates the appropriate sequence of functions at the hardware simulator 206.</u>
>
> The software kernel 212 functions as a communications controller when the target program 22 calls for interaction between the target microprocessor and the target circuitry.

Issue F:

Appellant argues that BU'948 is devoid of any teaching a programmable logic element temporarily suspending operations upon occurrence of a condition and then resuming operations from point of suspension when condition is serviced.

Issue argued in claims: 8, 9, 10, 18-20, 22-24, 28-29 and 32-34

Examiner's Response:

Examiner points to BU'948 Col.15 Lines 8-13, 36-67; Col.16 Lines 26-30 showing that processor emulator suspends operation upon occurrence of a condition. It is shown earlier that processor emulator has a programmable logic element (Fig.5 & 6). Interrupt condition in emulator suspends the operation which is serviced by hardware simulator and then target program 22 resumes on the hardware emulator.

BU'948 Col.15 Lines 8-13, 36-67; Col.16 Lines 26-30 respectively state:

Certain processor interface functions, such as breakpoint, an instruction trace, or memory reference to a location inside the emulator 202 (see FIG. 6), require no response from the target circuitry.

As previously discussed, the target circuitry may also initiate an event requiring interaction between the target microprocessor and the target circuitry. This is illustrated in FIG. 8B, where at the start 350, the target program 22 (see FIG. 6) is executing on the processor emulator 202. In step 352, the target circuitry, simulated on the hardware simulator 206 (see FIG. 6), signals an interrupt or other ***condition***, such as Power On, RESET, Bus Error, Address Error, Memory Fault, Non-Maskable Interrupt, and the like, requires interaction between the target circuitry and the target microprocessor. In step 354, the simulator interface, using the processor model shell 210, initiates a bus action corresponding to the particular interrupt or condition. This includes the generation of both timing and signal sequencing to simulate the activity of the target circuitry. In step 356, the processor model shell 210 communicates with the mapper 216. In step 360, the mapper 216 connects the hardware interrupt to a processor interface function. In step 362, the translator 214 translates the interface function to a call sequence or sequences. In step 364, the system invokes an emulator communications call.

In step 368, the system 200 transfers the call sequence to the emulator interrupt control. In step 370, the processor emulator 202 (see FIG. 6) **simulates a machine interrupt condition**. In step 372, the processor emulator 202 interrupts the normal flow of the target program 22 and transfers control to an interrupt handler in the target program. In step 374, the target program 22

processes the target interrupt using the target interrupt handler. ***Upon completion of the target interrupt processing, the system 200 returns control to the target program 22, which continues normal program processing.*** Thus, as illustrated in FIGS. 8A and 8B, the system 200 processes requests for interaction between the target microprocessor and the target circuitry regardless of which portion of the target system initiated the event requiring such interaction. Each processor emulator 202 maintains synchronization locally and executes its own target program 22 (see FIG. 6). If one processor emulator 202 is ***waiting*** for something to happen, this does not prevent another processor emulator from executing independently.

Also see Col.13 Line 35 Col.14 Line 5.

Issue G:

Appellant has argued that the physical processor of BU'948 is clearly different from the reprogrammable hardware emulator of the present invention.

Issue argued in claims: 10, 11, 21, 22, 24, 32 and 36

Examiner's Response:

Examiner would like to point out to Fig.5, which clearly shows that hardware emulator having a programmable logic element (as FPGA) (also Col.9 Lines 8-20). Appellant indication to Fig.6 having physical processor, which is the intended "target processor" (Col.8 Lines 29-30) is a hardware design that can also be downloaded to programmable logic element (as FPGA).

BU'948 Col.10 Lines 6-13, Col.8 Lines 29-30, Col.9 Lines 8-20 states*:

The present invention is embodied in a system 200 shown in FIG. 6. The system 200 includes a processor emulator 202, such as the microprocessor emulator 60 (see FIG. 4) manufactured by Applied Microsystems Corporation and others. The processor emulator 202 typically includes the target microprocessor itself as the microprocessor 76. However, the microprocessor 76 may be different from the target microprocessor and use additional components such as the FPGA to form a hardware circuit emulation of the target microprocessor. **The operation of the processor emulator 202 is similar 5 to that the of the conventional hardware emulators illustrated in FIG. 4 except for the operation of the control circuit 80 (see FIG. 4) and the control program 104 (see FIG. 5), which are replaced by the function of the control circuit 204.** The operation of the control circuit 204 in the system 200 will be discussed in detail below.

The processor emulator 60 includes a microprocessor 76, which is typically the target processor.

*Another form of hardware emulator, shown in FIG. 5, is a hardware circuit emulator 94. The hardware circuit emulator 94 uses reconfigurable circuitry 96, such as a field-programmable gate

> array (FPGA), to emulate target circuitry functions including the ASIC or custom IC. **Companies such as Quickturn and Aptix have developed hardware circuit emulators which allow hardware designs to be downloaded into the reconfigurable circuitry 96 and mounted on a board-like device**. The hardware circuit emulator 94 includes a processor chip 100 and a memory 102 containing the target program 22. The hardware circuit emulator 94 allows the target system to be tested as if it is already built.

Also See Col.13 Lines 55-59.

BU'948 teaching above clearly teaches that reprogrammable hardware emulator being used along with other teachings of the BU'948.

Appellant references to Col.15 Lines 8-13 in claim 10 specifically are addressed under Issue F, which now clearly identifies the section of the BU'948 teaching related to temporarily pausing operations of the reprogrammable logic element and then resuming them.

Issue H:

Appellant argues interpretation of Behavior Processor and states:

"The examiner cites BU'948, column 13, lines 55-60 as teaching a behavioral processor as a software kernel, control program and associated control circuitry surrounding the hardware emulator. The portion cited by the examiner simply does not teach a behavior processor. Instead the *[BU'948]* portion cited teaches certain conditions require the processor emulator **(hardware)** to communicate with hardware simulator **(software)** such as explicit memory references outside of the loaded target program address range, input/output operations, **interrupt handling**, and instruction dealing with explicit hardware functions."

Issue argued in claims: 13, 14, 15, 17-26

Examiner's Response:

As there is no clear definition presented for a behavior processor, Examiner would like to point to "**Summary of Claimed Subject Matter**" for definition, as presented in appeal brief (Pg.3).

"A novel behavior processor provides a unique architecture for implementing behavior applications, such as monitors, triggers, and memory server that is used in integrated circuit design verification. One embodiment of the present invention comprises a Behavior Processor that is integrated with a **reconfigurable computing (RCC) system** (i.e. host workstation containing a **software model** of at least a portion of the system design that is being emulated) and a **RCC hardware array** (i.e. an emulator containing the register transfer level (RTL) **hardware model**). With this configuration, behavioral aspects of a user's integrated circuit design and debug session are implemented in hardware to accelerate the design verification process."

Examiner asserts that the definition of behavior processor cited by BU'948, as restated by appellant, and the definition of behavior processor presented in the summary of claimed subject matter by the appellant have the same elements performing the same function. The interrupt handling is done to process the monitors and triggers initiated from hardware. Issue A is addressed separately in the beginning relating to the behavioral aspect of the current invention.

III In Response of Claims 4, 16, 31 and 35 under 35 USC 103(a)

Claims 4, 16, 31 and 35 have are argued to be allowable as they are dependent from claims 1, 13, 27 and 32.

Examiner respectfully points to issues A, B, C, F, G and H, which are presented for claims 1, 13, 27 and 32 as reasons for claims 4, 16, 31 and 35 to remain rejected.

The secondary reference IEEE1364 is argued as being devoid of any teaching or suggestion of using a reprogrammable logic element to model a behavioral function of user design.

Examiner has not relied upon IEEE1364 for support for this limitation. Further, the user's design (also in BU'948: Col.5 Lines 50-53) of integrated circuit is written in VHDL the IEEE industry standard presented in reference IEEE1364, therefore combination of BU'948 with IEEE1364 is obvious to one skilled in the art of programmable logic design (e.g. on FPGA). Please see motivation to combine.

IV In Response of Claims 12 and 37 under 35 USC 103(a)

Claims 12 and 37 have are argued to be allowable as they are dependent from claims 1 and 32.

Examiner respectfully points to issues A, B, and C which are presented for claims 1 and 32 as reasons for claims 12 and 37 to remain rejected.

Appellant has further performed piecemeal analysis to point out that ED1995 reference is devoid of any teaching or suggestion of using a reprogrammable logic element to model a behavioral function of user design or a test bench callback process responsive to behavioral function. ED1995 is commercial specification for a programmable logic element (e.g. FPGA) that teaches an FPGA running at 145 MHz to meet the claim limitations and motivation to combine is that BU'948 teaches use of an FPGA in Fig.5 for the hardware emulator.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Akash Saxena
Patent Examiner, GAU 2128
(571) 272-8351
Thursday, August 17, 2006

Conferees:

Fred Ferris
Primary Examiner, GAU 2128
Structural Design, Modeling, Simulation and Emulation
(571) 272-3778

Kamini S. Shah
Supervisory Patent Examiner, GAU 2128
Structural Design, Modeling, Simulation and Emulation

KAMINI SHAH
SUPERVISORY PATENT EXAMINER